

2. Funkce

FUNKCE

- VÝSLEDKEM je VÝPOČET JEDINÉ HODNOTY (u procedur následkem činností),

- DEKLARACE

- na vrcholu deklaracií proměnných

- SKLÁDÁ SE K HLAVICKY FCE A TĚLA FCE

function NAZEV_FCE (parametry funkce): typ funkce;
lokální deklarace; formální parametry
begin
postupy pro řešení
end;

- SPECIFIKACE FORMÁLNÍCH PARAMETRŮ

- stejný název a název jaký u procedury s parametry

- v TĚLE FUNKCE MUSÍ BYT PRÍRÁZOVACÍ PŘÍKAZ, KTERÝM
se přiřazuje hodnota místní funkce a její pravidlo
se definuje VÝSLEDNÁ FUNKČNÍ HODNOTA

- VYVOLÁNÍ (AKTIVACE)

- jménem FCE na místě, kde se očekává funkční hodnota
NAZEV_FCE (parametry funkce);

(př.) Použití funkce (podíl 2 cíbých čísel, dílce ≠ 0)

program POUZITI_FCE (input, output);
uses CRT;

var A, B: integer; C: real;

function PODIL (X, Y: integer): real; *typ funkce*

var Z: real; *formální parametry*
begin *rozšíření hodnoty*

Z := X/Y;

PODIL := Z → přiř. pěšky, na konci funkce

end;

{HLAVNÍ PROGRAM}

begin

close;

readln ('Když A,B, B<>0:');

readln (A,B);

C := PODIL (A,B); → rozšíření funkce, A,B sloučení parametry

readln (A,';',B,'=',C:1:2);

readln

end.

- kde je proměnná Z
function PODIL (X,Y: integer):
begin *typ funkce*
PODIL := X/Y.
end;

- je to proměnná C
(ale lepší s něj)

readln (A,B);
writeln (A,';',B,'=',
PODIL (A,B));
{je vyplňena až
na výpočet}

- ① napiš standardní funkci \tan jazyka Pascal novou funkcí tg .
 Napište ji v TG, kdežto počítá tangentu úhlu v radianech v radiánech
 a použijte ji v programu

$x \dots \text{v rad} (\text{rádiové číslo})$ $\text{tg } x = \frac{\sin x}{\cos x} \quad \sin(x), \cos(x) \dots \text{standardní funkce přidat v Pascalu}$ pomíná: skutečné x $y := \text{tg}(x)$	FORMÁLNÍ ... UHEL $\text{tg} := \sin(\text{UHEL}) / \cos(\text{UHEL})$
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------

```

program TANGENS (input, output);
uses CRT;
var X,Y: real;
function TG (UHEL: real): real;
begin
  TG := sin(UHEL)/cos(UHEL)
end;
{HLAVNÍ PROGRAM}
begin
  clscr;
  writeln ('*** FUNKCE TANGENS ***');
  writeln ('Kadý uhel v radianech:');
  readln (X);
  Y := TG(X);
  writeln ('Tangens uhel', X, 'rad je', Y:1:5, ':');
  readln
end.
  
```

- ② napiš funkci \cotg a použijte ji v programu (jako předchozí)
 Důvod

$x \dots \text{v rad}$ $\text{cotg } x = \frac{\cos x}{\sin x}$ pomíná: skutečné x $y := \text{cotg}(x)$	FORMÁLNÍ ... UHEL $\text{cotg} := \cos(\text{UHEL}) / \sin(\text{UHEL})$
----------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------

```

program KOTANGENS (input, output);
uses CRT;
var X,Y: real;
function COTG (UHEL: real): real;
begin
  COTG := cos(UHEL)/sin(UHEL)
end;
{HLAVNÍ PROGRAM}
begin
  clscr;
  writeln ('*** FUNKCE KOTANGENS ***');
  :
  Y := COTG (X);
  :
  
```

③ Napište funkci pro násobek m-faktoriálu. Použijte ji v programu pro násobek kombinačního čísla $\binom{m}{k} = \frac{m!}{(m-k)k!}$. Ověřte pro $\binom{8}{3}$.

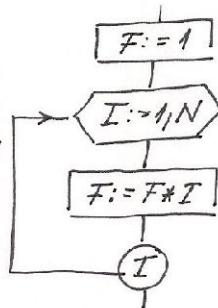
$$m! = m(m-1)(m-2) \cdots 3.2.1 \\ = \underbrace{1.2.3 \cdots (m-1)m}_{\text{SOUČIN PRÍROZENÝCH ČÍSEL} \leq N}$$

\Rightarrow CYKLUS FOR (s počtem darym počtem operací)

$$0! = 1 \quad 1! = 1$$

$$\binom{8}{3} = \frac{8!}{(8-3)!3!} = \frac{8 \cdot 7 \cdot 6 \cdot 5!}{5! \cdot 3!} = \frac{8 \cdot 7 \cdot 6}{3 \cdot 2 \cdot 1} = 56$$

$$\left[\text{nachází } \binom{8}{3} = \frac{8 \cdot 7 \cdot 6}{3 \cdot 2 \cdot 1} = 56 \quad \text{pozn. } \binom{8}{5} = \binom{8}{8-5} = \binom{8}{3} \quad \binom{m}{k} = \binom{m}{m-k} \right]$$

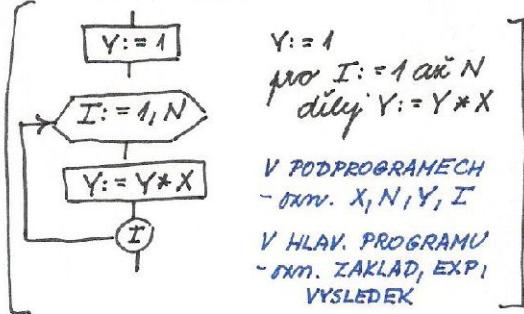


function FAKT(N: integer);
longint;
var F: longint;
I: integer;
begin
F := 1;
for I := 1 to N do
F := F * I;

program KOMBINACNI_CISLO(input, output);
uses CRT;
var N, K: integer;
C: longint;
function FAKT(N: integer): longint;
var I: integer;
F: longint;
begin
F := 1;
for I := 1 to N do F := F * I;
FAKT := F
end;
{HLAVNÍ PROGRAM}
begin
clrscr;
writeln('KOMBINACNÍ ČÍSLO');
writeln('Když N, K:');
readln(N, K);
C := FAKT(N) div (FAKT(N-K) * FAKT(K));
writeln('Kombinacní číslo je ', C, '!');
readln {repeat until kye pressed};
end.

- (4) Napište program pro nápočet mocniny zadaného čísla, kde exponent je předem uvedený číslo. Program měl využít také, kdy pro nápočet mocniny použijete
- proceduru bez parametrů
 - proceduru s parametry
 - funkci
(VÁSNEŘTE SI ROZdíLY)

$$3^5 = \underbrace{3 \cdot 3}_{MFX} \cdot \underbrace{3 \cdot 3}_{NOUČENÉ ČÍSLO} \cdot \underbrace{3}_{CYKLUS}$$



a) PROCEDURA BEZ PARAMETRŮ

program MOCNINA_CISLA (input, output);
uses CRT;
var ZAKLAD, VYSLEDEK: real;
EXP: integer;

procedure MOCNINA;
var X, Y: real;
N, I: integer;
begin
X := ZAKLAD; N := EXP;
Y := 1;
for I := 1 to N do Y := Y * X;
VYSLEDEK := Y
end;

{HLAVNÍ PROGRAM}
begin
elsece;
writeln ('Zadej základ
a přírodní exponent:');
readln (ZAKLAD, EXP);
MOCNINA;
writeln (EXP, '. mocnina
čísla', ZAKLAD, ', ji',
VYSLEDEK: 1:3, ',');

readln
end.

b) PROCEDURA S PARAMETRY

program MOCNINA_CISLA (input, output);
uses CRT;
var ZAKLAD, VYSLEDEK: real;
EXP: integer;
procedure MOCNINA (X: real; N: integer);
var Y: real;

var I: integer;
begin
Y := 1;
for I := 1 to N do Y := Y * X;
end;
{HLAVNÍ PROGRAM}
begin
:

readln (ZAKLAD, EXP);
MOCNINA (ZAKLAD, EXP, VYSLEDEK);
writeln (...
readln
end.

c) FUNKCE

program MOCNINA_CISLA (input, output);
uses CRT;
var ZAKLAD, VYSLEDEK: real;
EXP: integer;
function MOCNINA (X: real; N: integer):

real;
var I: integer;
Y: real;
begin
Y := 1;
for I := 1 to N do Y := Y * X;
MOCNINA := Y
end;
{HLAVNÍ PROGRAM}
begin
:

readln (ZAKLAD, EXP);
VYSLEDEK := MOCNINA (ZAKLAD, EXP);
writeln (...
readln
end.

- ⑤ Výškový program na kráčení zlomků, v kterém využijete funkci NSD pro určení největšího společného dělitel

- využijte Eukleidov algoritmus pro určení NSD

$$(20, 12) \rightarrow (8, 12) \rightarrow (8, 4) \rightarrow (4, 4) \quad \text{NSD}(20, 12) = 4$$

dokud $X \neq Y$ dělaj:

$$\begin{aligned} &\text{jste-li } X > Y, \text{ pak } X := X - Y \\ &\text{jinak } Y := Y - X \end{aligned}$$

$$\text{DEZ} := \text{NSD}(\text{CIT}, \text{JHEN})$$

- kráčení zlomku

$$\frac{20}{12} = \frac{\frac{20}{4}}{\frac{12}{4}} = \frac{5}{3}$$

$$\begin{aligned} \text{CIT_N} &:= \text{CIT} \bmod \text{DEZ} \\ \text{JHEN_N} &:= \text{JHEN} \bmod \text{DEZ} \end{aligned}$$

$\text{CIT} = 0$ mbož $\text{JHEN} = 0$
chybní zadání
finální

$$\text{DEZ} := \text{NSD}(\text{CIT}, \text{JHEN});$$

jedná se o DEZ > 1
urč. nový číslo jmen.
jedná se o $\text{DEZ} \leq 1$
atomické množství
- je v násled. krok

program KRACENI_ZLOMKEU (input, output);
uses CRT;

var CIT, JHEN, CIT_N, JHEN_N, DEZ: integer;

function NSD (X, Y: integer): integer;

begin

while $X \neq Y$ do

if $X > Y$ then $X := X - Y$ else $Y := Y - X$;

NSD := X

end;

{HLAVNÍ PROGRAM}

begin

elsecase;

writeln ('*** KRÁČENÍ ZLOMKEU ***');

writeln ('Zadej čitateli zlomku:');

readln (CIT);

writeln ('Zadej jmenovatele zlomku:');

readln (JHEN);

if CIT = 0 or JHEN = 0 then writeln ('chybní zadání!')

else

begin

DEZ := NSD (CIT, JHEN);

if DEZ > 1 then

begin

CIT_N := CIT mod DEZ;

JHEN_N := JHEN mod DEZ;

writeln (CIT, ':', JHEN, '=', CIT_N, ':', JHEN_N)

end

else writeln ('Zlomek', CIT, ':', JHEN, 'je v následujícím
stavu.')

end;

readln

end.